面向ONSET实时数据处理的图像选帧GPU技术实现

李力[1]，邓辉[1*]，李臻[3]，梅盈[2]，戴伟[12]，杨秋萍[12]，王锋[12]
1 昆明理工大学云南省计算机技术应用重点实验室 云南 昆明 650500
2 中国科学院云南天文台 云南 昆明 650011
3 南京大学天文与空间科学学院 江苏 南京 210000

摘要：光学和近红外太阳爆发监测望远镜每天可以获得大量的太阳图像数据，对这些观测数据进行实时选帧处理，一方面可以减轻存储压力，另一方面也可以提高后续图像重建的质量。针对实时观测过程中的选帧要求，设计并实现了一套基于图形处理器的图像选帧实时处理模块，当前的模块已经实现了平均梯度法和谱比法选帧两种算法的高速并行处理。对模块的实现进行了细致的讨论，并比较了两种选帧方法的加速比。实验表明，该模块运行稳定可靠；从执行效率来看，针对近全日面图像的选帧总体执行时间最快为1.2s，比原有串行实现提升了7倍；局部面图最快为0.7s，平均提升了5倍。整体模块的实现与当前性能已经可以满足实时观测与处理的要求。

关键词：实时选帧；光学和近红外太阳爆发监测望远镜；图形处理器；实时选帧
中图分类号：TP311.1　　文献标识码：A　　文章编号：1672-7673

光学和近红外太阳爆发监测望远镜(ONSET)可实现3个通道共4个波段高时间和高空间分辨率的太阳观测，视场可全日面和局部面切换，是一台对发展我国太阳物理观测研究和空间天气监测具有重大意义的地基太阳望远镜[1]。

和所有地基望远镜一样,为克服大气湍流的影响，获得接近或者达到衍射极限的图像[2]，光学和近红外太阳爆发监测望远镜拟采用高分辨图像重建方法来进行实时数据处理，处理过程包括：针对一组原始短曝光图进行选帧和重建。重建之前进行选帧处理不仅可以提高后续重建的图像质量，同时在观测中只保留质量较好的图像，也能减轻观测原始数据的存储压力。

显然，图像质量评价是选帧技术的关键组成部分。由于在地面观测中很难得到观测目标的一帧原始无偏差图像作为评价参考，因此所有的全参考型和半参考型图像质量评价方法如基于结构相似度的方法[3]、均方误差和峰值信噪比方法[4]均不能用于天文观测数据的选帧处理。而在无参考评价方法中，平均梯度法[5]反映了图像灰度变化率的大小，并且图像的细节反差变化速率衡量的重要标准就是图像的平均梯度；谱比法[6-7]在太阳高分辨图像选帧中得到运用，对于实验图像的要求是图像要达到足够多的数量而且这些图像在拍摄时间上连续。

本文的研究工作是针对ONSET实时观测过程中的选帧要求，设计并实现了一套基于图形处理器的选帧实时处理模块，当前模块实现了平均梯度法和谱比法的高速并行处理。实验结果表明，本模块运行稳定可靠，整体实现与性能已经可以满足光学和近红外太阳爆发监测望远镜实时观测与处理的要求。

1. 图形处理器并行技术

图形处理器起初是向中央处理器提供一些基本的操作。但随着图形处理器的发展，已经成为具有高并行度的高性能计算处理器。统一计算架构(Compute Unified Device Architecture, CUDA)，是英伟达为图形处理器增加的一个提高通用计算能力且易用的编程接口。程序员通过使用C语言在CUDA架构下编写程序，就可以在英伟达的图形处理器下运行①。CUDA工具箱(CUDA Toolkit)包含两个重要的工具库，一个是优化后的快速傅里叶变换(Fast Fourier Transform)库，CUFFT；另一个是线性代数函数(Linear Algebra Routine)库，其中包含了基本线性代数子程序(Basic Linear Algebra Subprograms, BLAS)，CUBLAS。

在中央处理器主机和图形处理器设备的数据交互中，通常数据首先从主机复制到设备内存中，然后由设备中的线程完成对数据预定的操作，并将最后的结果拷贝返回主机。合理分配图形处理器设备中的网格、线程块、每个块的线程数量，采用众多线程尽可能的达到高并行度，就可以利用图形处理器强大的计算吞吐量，达到选帧算法的高性能处理的目

基金项目：国家重点基础研究发展计划（973计划）（2016YFE0100300）；国家自然科学基金（U1231205，U1531132，U1631129，11403009，11463003）资助；云南省自然科学基金；云南省应用基础研究项目）
作者简介：李力，男，硕士研究生，研究方向：计算机应用技术，Email:lili@cnlab.net
通讯作者：邓辉，女，教授，研究方向：分布式计算与科学数据处理、云计算，Email:dh@cnlab.net

的。文[8]研究了太阳高分辨图像重建算法在图形处理器上的实现。文[6]实现了基于图形处理器并行计算的 Level1 级重建程序，使程序在不改变原有处理方式的情况下将重建速度提高近 30 倍，达到了准实时重建的效果。文[9]在 CUDA 环境下实现了一个 Hα 全日面云污染实时识别和修复系统。

以上分析可见，图形处理器并行技术已经在天文应用开发中有诸多应用，为开发光学和近红外太阳爆发监测望远镜观测数据中的实时选帧功能提供了思路，但是针对光学和近红外太阳爆发监测望远镜具体产出的数据特点及选帧要求，需要进一步研究开发能够适合光学和近红外太阳爆发监测望远镜选帧算法的并行实现技术，从而满足选帧的实时性要求。

2. 选帧实时处理模块的设计

光学和近红外太阳爆发监测望远镜可实现多通道多波段的观测，不同的数据类型具有不同的图像特点，不同波段的图像有可能采用不同的选帧算法。为了满足观测过程中的实时选帧要求，有必要设计并实现一套选帧实时处理模块。

一般来说，在光学和近红外太阳爆发监测望远镜观测过程中，数据选帧首先是获取一组待选帧图像，然后通过图像质量评价方法计算每帧图像的评价值，依据评价值按照预期的选帧比例进行挑选，最后将优选帧和剔除帧分开存储，以完成整个选帧过程。

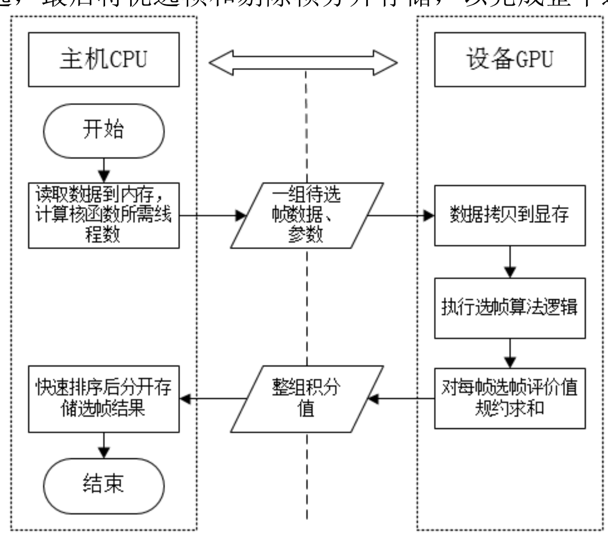


图 1 并行实现选帧算法流程图

Fig.1 A flow chart of frame selection algorithm for parallel implementation

在具体的实现中，如果考虑到中央处理器和图形处理器的异构特点，设计的选帧算法并行实现模块流程图如图 1。

在实现中针对选帧算法设计的本模块的步骤主要包括以下几个功能流程：

（1）读取一组光学和近红外太阳爆发监测望远镜斑点图（如 100 帧）到主机内存；

（2）申请图形处理器设备内存，拷贝到该组数据；

（3）根据每帧图像的大小计算设备执行核函数所需线程数；

（4）在图形处理器中执行具体选帧算法逻辑（后面的实验中具体实现了平均梯度法和谱比法）；

（5）对图形处理器计算的结果进行规约求和，结果拷贝回主机内存；

（6）在主机中，根据所求每帧评价值，对该组数据进行快速排序；

（7）根据排序结果将好帧（如评价值前 70%）与坏帧分开存放到不同文件目录。

表 1 选帧实时处理模块示意代码

Table.1 Illustrate code of frame selection real-time processing module

选帧实时处理模块

`int main(){`
`  //读取待选帧数据`
`  for(i=0;i<sumImg;i++){`
`    ReadFitsImg(); //使用 cfitsio 库读取到主机内存`
`  }`

① http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

```
    cudaDeviceProp prop;
    KernelMaxThreadsPerBloc=prop.maxThreadsPerBlock; //获取设备每块的最大线程数
    ImgFrameSelection(帧数据,参数); //调用选帧处理方法
}
void ImgFrameSelection(){
    cudaMalloc((void**)&devAllImg,sizeof(float)*NX*NY*sumImg)); //申请设备内存
    cudaMemcpy(devAllImg,allImg,sizeof(float)*NX*NY*sumImg,cudaMemcpyHostToDevice)); //拷贝数据
    (NX*NY+ KernelMaxThreadsPerBloc-1)/ KernelMaxThreadsPerBloc; //计算所需线程块
    KernelFrameSelection<<<线程块,每块线程数>>>(); //具体选帧算法逻辑
    arraySumGpu(); //规约求和
    //将积分值拷贝回主机并关联文件名
    for(i=0;i<sumImg;i++){
      cudaMemcpy(&seqSub[i][0],SumResult+i,sizeof(float),cudaMemcpyDeviceToHost);
      seqSub[i][1]=i;
    }
    sortImg(); //快速排序
    frameSelectionResults(); //完成选帧
}
```

示意代码如表 1。为今后光学和近红外太阳爆发监测望远镜的发展，充分考虑到其科学目标的变化、观测终端的变化，可适应待选帧图像特点，具有较强的开放性和可扩展性在当前的模块实现中，在给图形处理器分配线程数时，主要依据待处理每帧图像的尺寸，使每个像素点可以分配到一个线程，每个线程对应处理一个灰度值的计算，整体来看每帧图像的所有像素点是并行计算和处理的。其中，在图形处理器中执行的选帧算法逻辑部分是选帧的关键，在这个环节，可以根据待选帧图像的特点和选帧要求，灵活地进行相应选帧算法的选取，然后集成到未来的光学和近红外太阳爆发监测望远镜数据处理系统中。

总体来看，模块实现尽可能地优化了选帧的执行效率，具有较高的并行度。在对每帧选帧评价值规约求和时，利用图形处理器中的共享内存实现并行规约思想。首先取得当前图形处理器中每个线程块所能分配的最大线程数，如果所需计算的每帧像素数小于单个线程块的最大线程数，则分配一个线程块，否则依据每帧像素数计算需要分配多少个线程块在计算每个线程块内的求和时，由于共享内存缓存中的偏移就等于线程索引，因此首先申请共享内存缓冲区所能分配的最大值，然后将线程块内的每个线程计算的值保存在这个缓冲区，最后对其进行规约迭代。每个线程块都按照这样的策略同时运算，就能高效得到整组选帧数据的评价值积分结果。

3. 基于图形处理器的选帧算法实现

结合光学和近红外太阳爆发监测望远镜观测数据和选帧算法的特点，目前实现了一种空域方法和一种频域方法即平均梯度法和谱比法。

3.1 平均梯度

平均梯度方法可以体现图像中物体的边缘和细节的强度，求平均梯度值的计算公式如下：

其中，M 和 N 分别代表图像的总行数和总列数；为图像中第行中第列的灰度值，根号里面代表了 F 点与其相邻的下方点和右侧点的灰度值之差的平方和。

从（1）式可以看出，每帧图像的所有像素点都要参与计算，必然带来很大的计算量。在平均梯度法不使用图形处理器并行加速的情况下，可以通过只计算图像的部分区域提升计算速度。因此这里对其进行简化，即只计算缩小一倍的中心区域。对于一组每帧尺寸为 2560*2160 的太阳近全日面图，选取中心 1280*1080 大小的区域进行计算；对于一组每帧尺寸为 1700*1700 的太阳局部面图，选取中心 850*850 大小的区域进行计算。

① http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

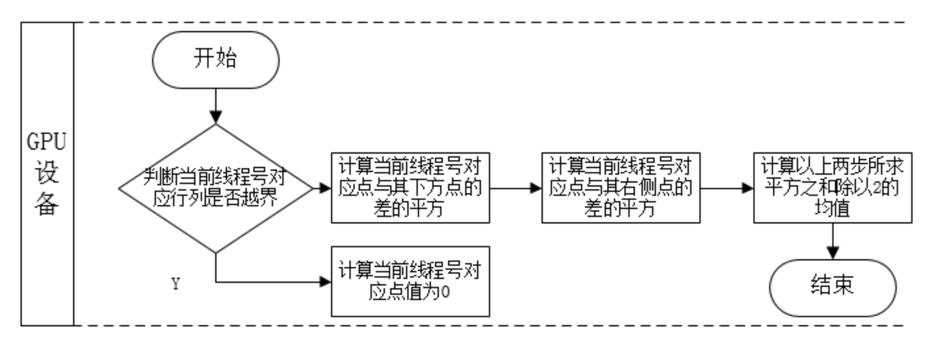


图 2 并行实现平均梯度选帧算法流程图

Fig.2 A flow chart of average gradient frame selection algorithm for parallel implementation

平均梯度法使用图形处理器并行加速的情况下，针对每帧图像的全域计算平均梯度值，在算法的并行实现中，每个线程主要参与计算以下几步，如图 2。关键核函数的示意代码，如表 2。

（1）判断当前行列是否越界，计算当前行列对应的像素点与其下方点的灰度值之差的平方；

（2）计算当前行列对应的像素点与其右侧点的灰度值之差的平方；

（3）将上述两个平方之和除以 2 再求均值。

表 2 平均梯度选帧算法关键核函数示意代码

Table.2 Illustrate code of average gradient frame selection algorithm for key kernel function

平均梯度选帧算法关键核函数

```
__global__ void GetGradient(float *devAllImg, int sumImg, int NX, int NY, float **Gradient){
  int tid = threadIdx.x + blockIdx.x * blockDim.x;//线程号
  if(tid<NX*NY){
    row = tid / NY ;//线程对应原图像中的行号
    column = tid - row * NY ;//线程对应原图像中的列号
    //每个线程对应计算所有图的相同位置，f0、f1、f2 为当前点与其相邻点灰度值
    for (int k = 0; k < sumImg; k++) {
       if(row<NX-1 && column<NY-1){
          //行方向的梯度值
          rowDirection = (fi1 - f0)*(fi1 - f0);
        //列方向的梯度值
          columnDirection = (fj1 - f0)*(fj1 - f0);
          //每个点的梯度值
        Gradient[k][(row)*NY+(column)] = sqrt( ( columnDirection + rowDirection) / 2) ;
          }
      }
    }
}
```

3.2 谱比法

谱比法选帧的主要原理是以目标区域功率谱的强弱进行选帧评价。其主要思路是将每帧图像的功率谱与该组图像序列中所有的图像的平均功率谱进行归一化处理，将相关目标的量消除，之后在目标图像的频段上进行积分，以积分值的大小作为选帧比较的依据，积分值大的图像认为质量较好。

谱比法需要使用多次傅里叶变换，在不使用图形处理器并行加速的情况下，只能采用 FFTW 库计算快速傅里叶变换的操作，由于傅里叶变换较为耗时，对于串行和并行情况下采取截取中心区域的方法，即对于近全日面图，确定起始行为 540，起始列为 640，终止行为 1619，终止列为 1919；对于局部面图，确定起始行列为 425，终止行列为 1274。

① http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

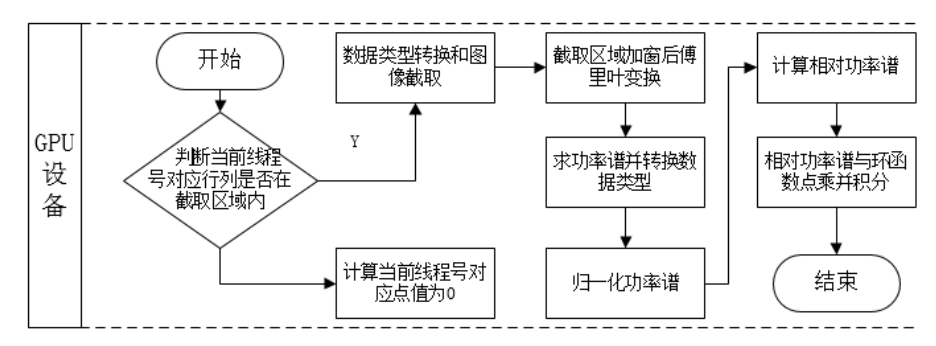


图 3 并行实现谱比法选帧算法流程图
Fig.3 A flow chart of spectral ratio frame selection algorithm for parallel implementation

采用图形处理器并行加速的情况下，如图 3，计算谱比选帧评价值的核函数的处理流程如下：

（1）在截取区域内，每个线程将对应的每个点的数据由实数转换为复数，并且在转换图像数据类型时对每帧大图进行截取操作；

（2）将截取区域与窗函数点乘，即为对截取区域加窗，同时进行复数到复数的傅里叶变换；

（3）计算上一步傅里叶变换后的值，获取功率谱，同时转换数据类型为实数；

（4）功率谱除以零频得到归一化的功率谱，同时将零频值置为 1；

（5）得到该组图像序列的归一化功率谱的均值，计算相对功率谱；

（6）相对功率谱与环函数点乘并积分，得到在目标频段上的选帧评价值。在并行算法中采用 CUFFT 库代替 FFTW 库执行快速傅里叶变换的操作。为使每个线程对应于图像中的一个像素点，同时考虑到并行算法中线程数分配的最优化原则，这里根据截取区域确定核函数分配的线程数。出于优化考虑，这里采用 CUBLAS 库函数 cublasSscal 求平均功率谱，采用 cublasSasum 计算在目标频段上的积分。

4. 实验与性能对比

为了验证本文的算法实现，进行了全面的实验。测试数据共 5 组，包括 1 组光学和近红外太阳爆发监测望远镜观测的 Hα 的太阳近全日面像（每帧尺寸 2560*2160，占用空间 11MB）和 4 组具有明显太阳活动的局部像（前两组为 Hα 图，后两组为白光图。每帧尺寸 1700*1700，占用空间 6MB）。在选帧目标上，对每组 100 帧的近全日面短曝光图挑选出来最好的 70 帧，对每组 70 帧的局部像挑选 50 帧。

4.1 平均梯度

在平均梯度法不使用选帧实时处理模块图形处理器并行加速的情况下，通过只计算图像的部分区域可以提升计算速度。如图 4，在这种提升选帧效率的方式下，部分区域串行时间明显低于全图区域串行时间，有平均 3.7 倍的提升。但是，这种方式换来的提升会损失部分有效的图像质量评价信息，势必会影响选帧的准确度。

图 4 平均梯度算法时间比较图
Fig.4 A diagram of average gradient algorithm for time comparison

在平均梯度法使用选帧实时处理模块 GPU 并行加速的情况下，针对每帧图像的全域计算平均梯度值。如图 4 所示，针对全图计算的并行算法仍然较只计算部分区域的串行算法速度优势明显，是全图区域串行时间的 6-8 倍。实验表明，平均梯度法的选帧实时处理模块运行稳定可靠，实现的并行算法的执行效率明显优于原有串行情况下的执行效率，有效性显著。

4.2 谱比法

如图 5，在谱比法使用选帧实时处理模块图形处理器并行加速的情况下，对于相同的处理区域，并行实现谱比算法的执行效率明显优于串行情况下的执行效率。针对近全日面图像的选帧总体执行时间最快为 1.2s，比原有串行实现下提升了 7 倍；局部面图最快为 0.7s，平均提升了 5 倍。实验表明，集成谱比法后，本文选帧实时处理模块运行稳定可靠，

① http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

选帧效率提升明显。

图 5 谱比法算法时间比较图
Fig.5 A diagram of spectral ratio algorithm for time comparison

5. 结束语

本文提出并实现一套基于图形处理器的选帧实时处理模块，对模块的设计和实现进行了细致的讨论，并已经在图形处理器下实现了平均梯度法和谱比法的高速并行处理。实验比较得出，模块取得了较好的加速比，性能已经可以满足光学和近红外太阳爆发监测望远镜实时观测与处理的要求，为实时高分辨图像重建打下了扎实的基础。

随着光学和近红外太阳爆发监测望远镜数据处理的需要和不同科学研究工作的需要，在现有模块下进一步发展，研究适合光学和近红外太阳爆发监测望远镜不同类型特点的选帧算法，并实现和集成到未来的光学和近红外太阳爆发监测望远镜数据处理系统是下一步研究的内容。

参考文献

[1] 李臻. 太阳低层大气中小尺度活动的观测和研究[D]. 南京:南京大学, 2016.

[2] 刘长玉. 选帧技术在太阳高分辨成像中的应用[D]. 昆明:中国科学院云南天文台, 2013.

[3] Wang Z, Bovik A C, Sheikh H R, et al. Image quality assessment：from error visibility to structural similarity[J].IEEE Transaction on Image Processing，2004，13(4):600-612

[4] 郑艳芳. 太阳高分辨数据评价与喷流/日浪的观测研究[D]. 昆明:中国科学院云南天文台, 2014.

[5] 赵青, 何建华, 温鹏. 基于平均梯度和方向对比度的图像融合方法[J]. 计算机工程与应用, 2012, 48(24): 165－168.
Zhao Qing， He Jianhua， Wen Peng. Image fusion method based on average grads and wavelet contrast[J].Computer Engineering and Applications，2012，48（24）：165-168.

[6] 施正, 向永源, 邓辉, 等. 一米真空太阳望远镜 Level 1 级图像选帧的 GPU 实现[J]. 科学通报, 2015, 60(15):1408－1413.
Shi Zheng，Xiang Yongyuan，Deng Hui，et al. The frame selection for New Vacuum Solar Telescope Level 1 data processing based on GPU [J]. Chinese Science Bulletin,2015,60(15)：1408–1413.

[7] 刘长玉, 向永源, 金振宇. 太阳色球图像的选帧处理[J]. 天文研究与技术—国家天文台台刊, 2014, 11(2):140－144.
Liu Changyu， Xiang Yongyuan， Jin Zhengyu. Image processing with frame selection for Hα solar chromosphere images[J]. Astronomical Research and Technology—Publications of National Astronomical Observatories of China，2014，11(2)：140–144.

[8] 向永源. 太阳高分辨高速重建算法的研究[D]. 昆明:中国科学院云南天文台, 2016.

[9] 张能维, 杨云飞, 李冉阳, 等. Hα 全日面云污染实时识别和修复系统[J].天文研究与技术,2016,13(2)：242-249.
Zhang Nengwei，Yang Yunfei，Li Ranyang，et al. A real-time image processing system for detecting and removing cloud shadows on Hα full-disk solar [J].Astronomical Research and Technology，2016，13(2)：242-249.

① http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

Implementation on Image Frame Selection of GPU for ONSET Real-time Data Processing

Li li[1], Deng hui[1*], Li zhen[3], Mei ying[2], Dai wei[12], Yang qiuping[12], Wang feng[12]
1Yunnan Key Laboratory of Computer Technology Application, Kunming University of Science and Technology, Kunming City, Yunnan Province, 650500
2Yunnan Observatory of Chinese Academy of Sciences, Kunming City, Yunnan Province, 650011
3School of Astronomy & Space Science, Nanjing University, Nanjing City, Jiangsu Province, 210000

Abstract: Optical and Near-infrared Solar Eruption Tracer(ONSET) can get a lot of sun image data every day. By real-time frame selection to these observation data, on the one hand can reduce the storage pressure, extend the observation time, the other hand can also improve the quality of subsequent image reconstruction. In this paper, a set of real-time processing image frame selection module based on GPU is designed and implemented for the frame selection in real-time observation process, and it has achieved high-speed parallel processing of gradient method and spectral ratio method in the module. This paper has discussed the implementation of the module in detail, and compared the speed of the two frame selection methods by experiment. Experimental results have shown that the module is stable and reliable. From the implementation efficiency, the best overall execution time of frame selection for nearly full view image is 1.2 seconds, and that achieve an elevated seven times more than the original serial implementation. For partial view image is 0.7 seconds, and that makes an average increase of five times. Implementation and current performance of the overall module have been able to meet the requirements of real-time observation and processing.

Key words: Real-time frame selection; ONSET; GPU

① http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html